Transitioning a Flexible and Scalable Satellite Ground Station Observation Network (GSON) Framework to an Operational Environment

LOUIS NGUYEN
NASA Langley Research Center, Hampton, Virginia

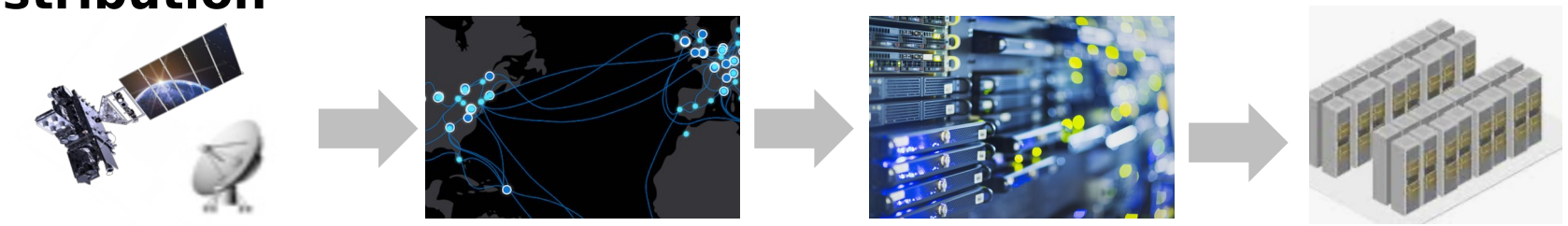
THAD CHEE AND ANDREI A. VAKHNIN
Science Systems and Applications Inc. Hampton, Virginia

STEWART ALAN SUTTON, ANDREW MELNICK, DONALD J. BOUCHER AND JOHN C. MERGO
The Aerospace Corp., El Segundo, California

AMS 103rd Annual Meeting
13th Conference on Transition of Research to Operations
Colorado Convention Center, Denver, Colorado
January 8-12, 2023

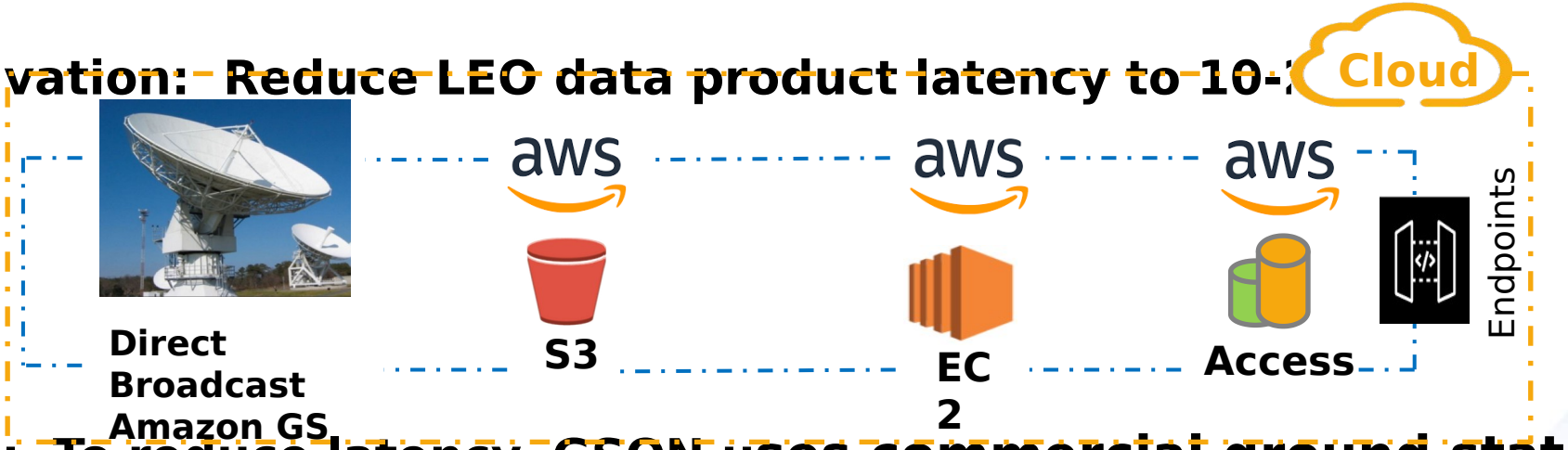
GSON: Introduction and Motivation

- **Problem: LEO data latency (3-6 hours) poses a significant impact on data optimal use due to data latency at data acquisition, processing & distribution**



Reception at Svalbard GS | **Transfer to Data Center** | **Data Center Processing** | **Access and Distribution**

- **Motivation: Reduce LEO data product latency to 10-2**



- **How: To reduce latency, GSON uses commercial ground station and cloud provider to receive and process direct broadcast data from EOS satellites**

Genesis of the project

- **AWS Ground Station debuted in 2018 but access became available to government teams in 2020**
- **A number of groups (NASA, NOAA, DoD) did similar explorations**
- **Each group ended up exploring different aspects**
  - NASA/LaRC developed GSON:
    - focuses on high speed processing of single satellite overpasses
    - A key goal was the flexibility of processing
  - Aero group worked on large volumes of stored mission data (SMD)
  - The requirements were different:
    - Single overpass focuses on speed and flexibility
    - SMD processing focuses on accessing large amounts of compute

Operations / Merging Resources

- **After operating respective systems over two years, new technologies evolved from lessons learned in the operation:**
- **Aero Group focused on scaling and reliable script operation:**
  - An AWS cluster provides scalable access to highly available compute and storage resources
- **GSON became part of NASA NOS-T project resulting in a focus on application processing flexibility and interoperability:**
  - Message Passing architecture was integrated
  - Processing engine was developed to allow easy modification and coordination of different processing tasks

Lessons Learned -> Requirements

- **GSON**
  - Containerization is key to flexible and rapid development and deployment
  - Containerization removes library version issues and isolates the operating pieces from operating system specifics. The result is simpler deployment requirements
  - Hybrid containers solve the huge container problem – lookup tables, reference files are stored in network storage resulting in “smaller” containers. This introduces complexity – more about this on the next slide
- **AERO**
  - Code update, deployment and operation – Software Factory
  - Code synchronization will always be an issue – Panopticon
  - Authentication is/was a hassle – Anarchy
  - Cluster implementation was hard to track/control – Immature

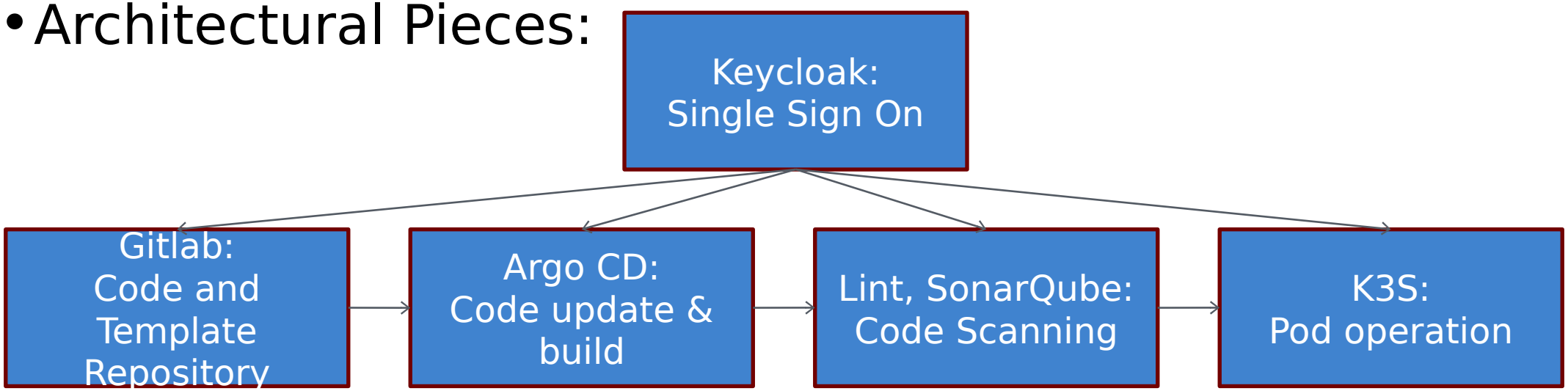
Scalable Environment - Requirements Based on Lessons Learned

- **Aero – Functional requirements and the systems that provide them:**
  - Code scanning – Lint, SonarQube
  - Single Sign on - Keycloak
  - Code Repository – Gitlab
  - CI/CD – FluxCD/ ArgoCD
  - Deployment – Ansible, Terraform

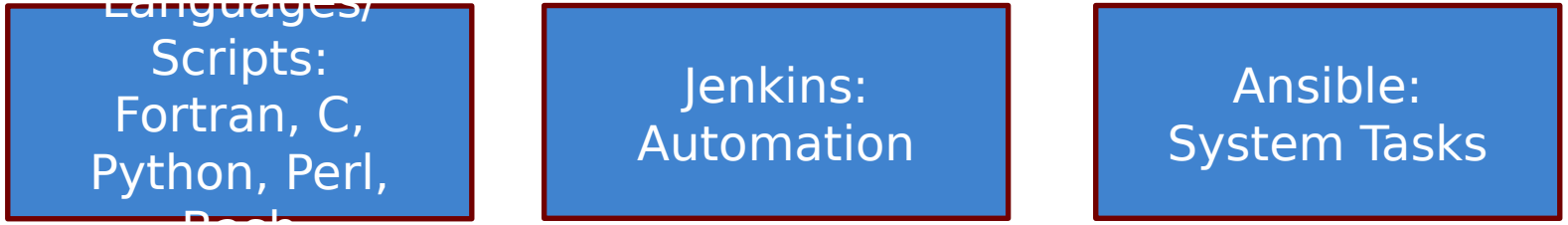
- **GSON –As an application, GSON has simpler requirements:**
  - Hybrid Container Support : Network storage (NFS / SMB)
  - SQL database
  - Network access to inputs

Software Factory Architecture

- Architectural Pieces:



- **Supporting Technologies**

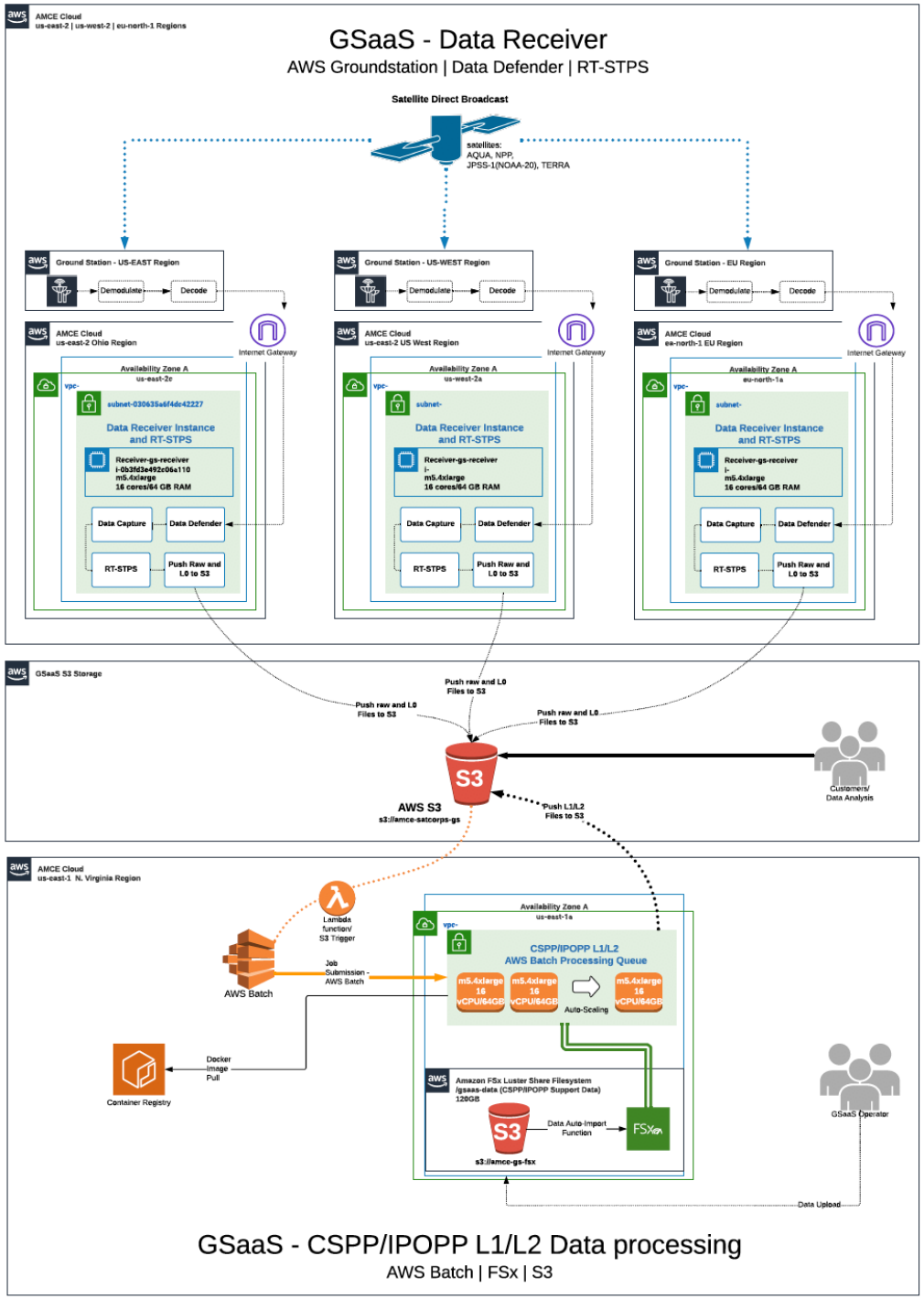


Processing Cluster Tools / Jobs

- **Deployment:**
  - Ansible *(https://www.ansible.com/)*
  - Terraform *(https://www.terraform.io/)*
- **Code scanning:**
  - Lint
  - SonarQube *(https://www.sonarsource.com/products/sonarqube/)*
- **Single Sign on:**
  - Keycloak *(https://www.keycloak.org/)*
- **Code Repository:**
  - Gitlab *(https://about.gitlab.com/)*
- **CI/CD:**
  - FluxCD/Helm (eol)
  - ArgoCD *(https://argoproj.github.io/cd/)*

GSON Implementation within Software Factory

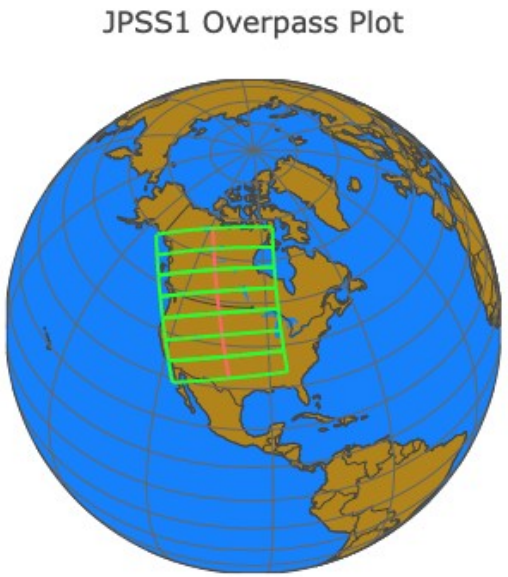
- **This is a ”Virtualized Ground Segment” Implementation (ground station 2.0)**
- **Network lookup storage**
- **Processing containers**
- **S3 storage**
- **Not Shown:**
  - External processing
  - External Data Sources (Azure, NOAA via AWS Open Data Registry

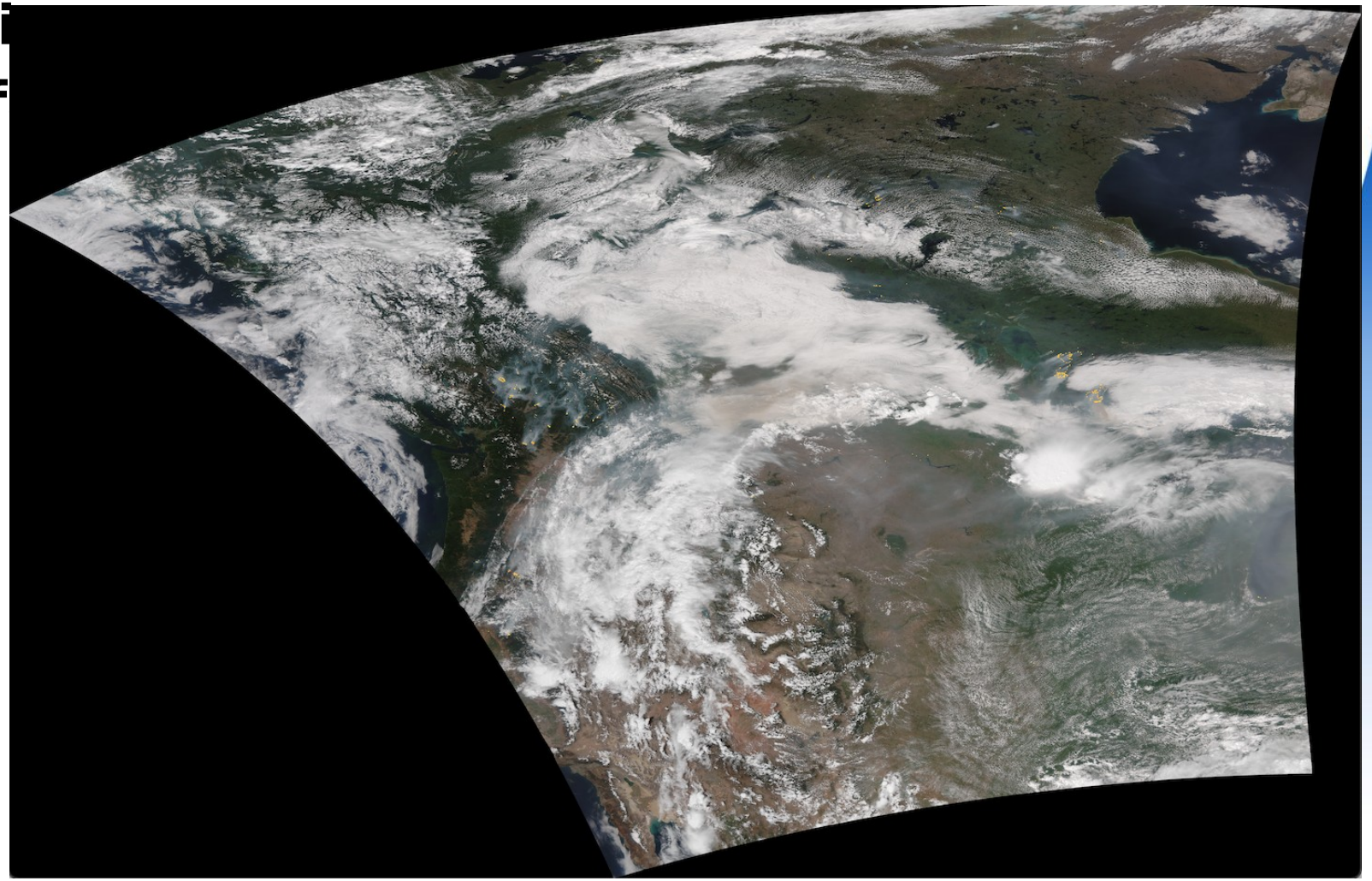


GSON Implementation

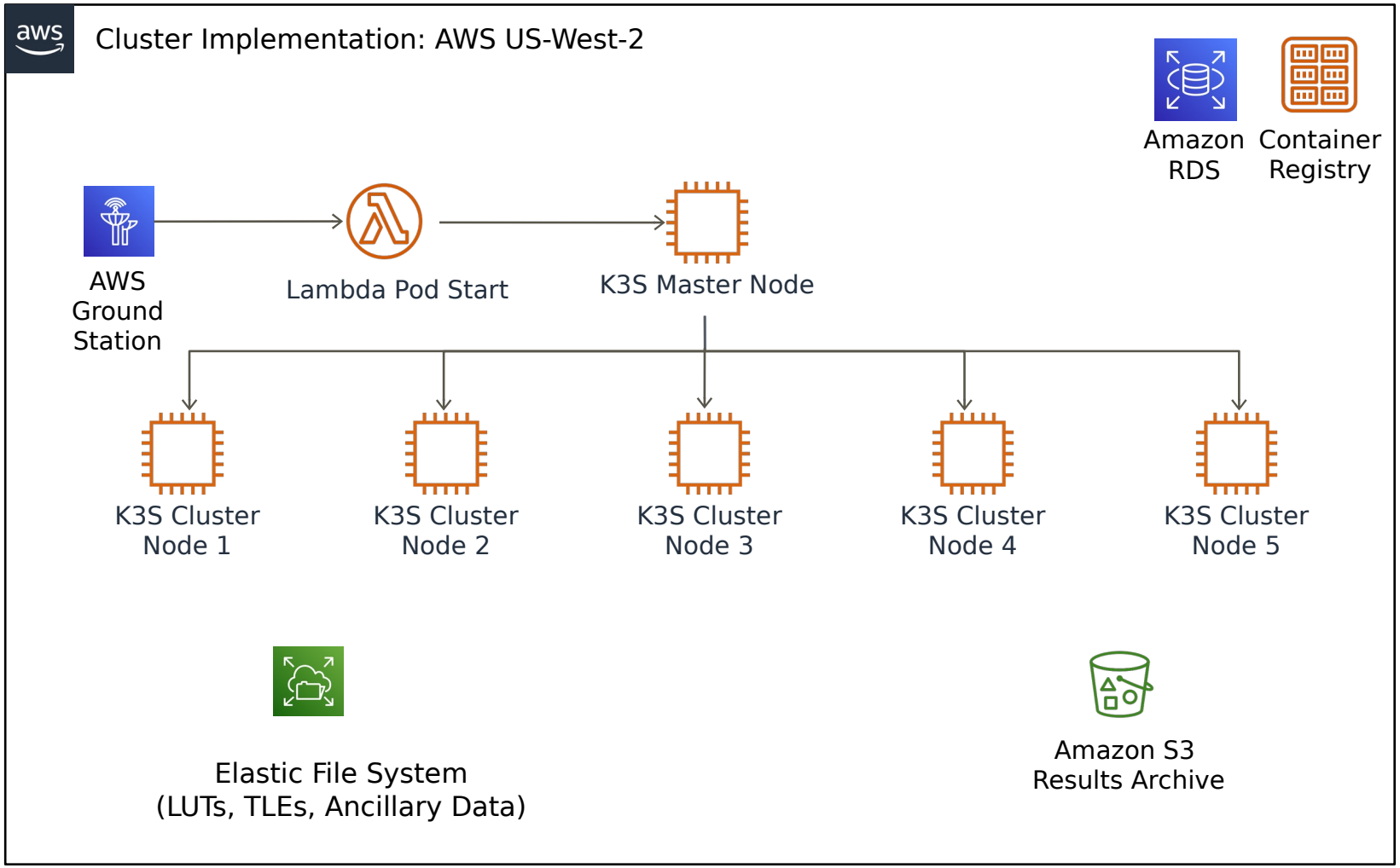
Application Technologi

- **Ordering API / Interf**
- **Parallel Processing Engine**
- **Messaging System**





Implementation Details

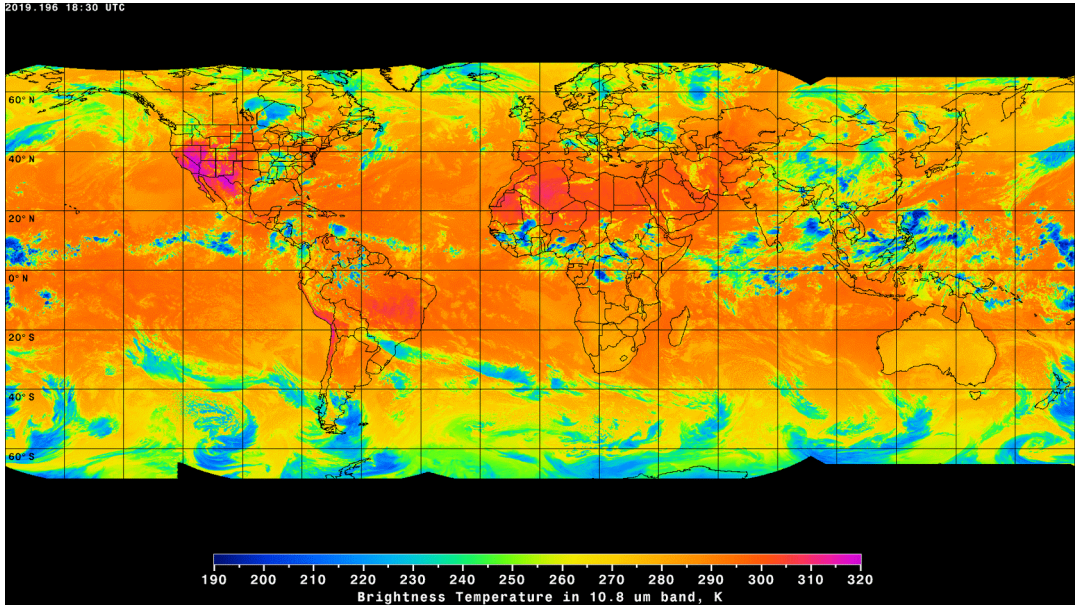


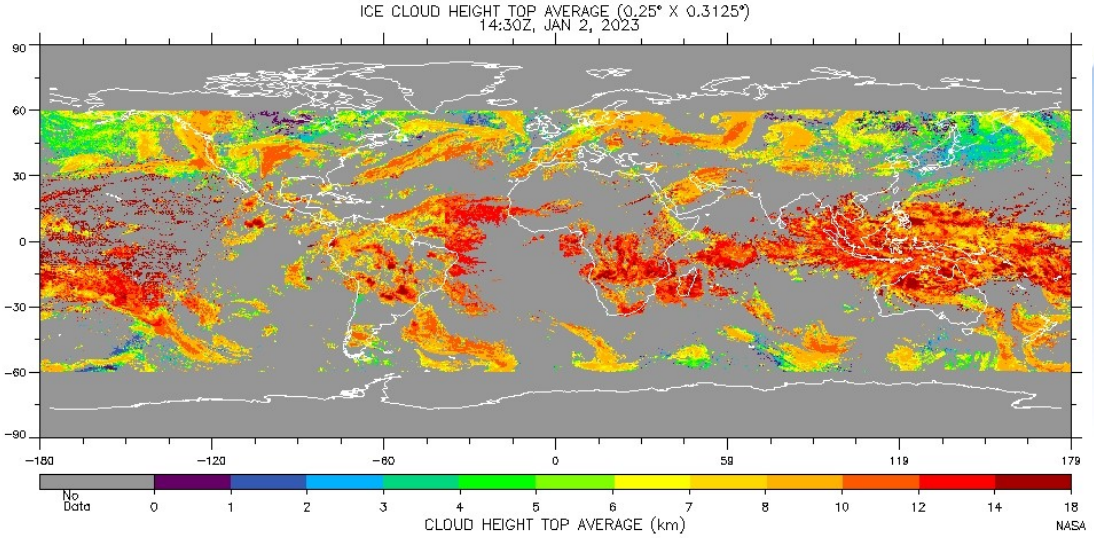
- **K3S Master Node:**
  - t3.small
- **K3S Cluster Nodes:**
  - t3.2xlarge
  - 8GB EBS (system)
  - 20GB cluster storage
- **EFS Network Storage:**
  - Unlimited storage size
- **S3 Results Archive:**
  - Distribution of results
  - Archive of results
- **Database (RDS):**
  - Order tracking
  - Processing Status
  - Metadata Storage

Current System Status

- **We are operating the cluster as a software factory concept to produce:**
  - Direct Broadcast L1 & L2 files
  - Cloud Products
  - Global Cloud Composite (GCC) products

- **The cluster software factory concept allows us to operate not only GSON but also SatCORPS and SatCORPS/GCC processing**





Going Forward, New Research

- **Cluster: Aero group is focusing on a “recipe” deployment of not only K3S but also adding the other pieces so that the cluster described can be deployed in less than an hour – assuming you have an AWS account.**

- **GSON: Multi system inputs – We can process Direct Broadcast from both AWS Ground Station and Microsoft Azure in real time. We are working on adding NOAA’s JPSS-1 and 2 from the AWS Open Data Registry and archive processing from NOAA CLASS.**

Closing

- **Thank you to all the team members for supplying information for this presentation!**

- **The multi-disciplinary nature of the project means that it’s a challenge to explain all of it without input from each specialty team member.**

Questions / Comments

Transitioning a Flexible and Scalable Satellite Ground Station Observation Network (GSON) Framework to an Operational Environment

- LOUIS NGUYEN (L.NGUYEN@NASA.GOV)
- *NASA Langley Research Center, Hampton, Virginia*
- THAD CHEE AND ANDREI A. VAKHNIN
- *Science Systems and Applications Inc. Hampton, Virginia*
- STEWART ALAN SUTTON, ANDREW MELNICK, DONALD J. BOUCHER AND JOHN C. MERGO
- *The Aerospace Corp., El Segundo, California*